

How to be a 1337 h4ck3r: Git + Linux

An introduction to Git, version control, and Linux

by Samsara Counts, Pat Cody, and Joseph Schiarizzi

Slides adapted from Neel Shah and Phil Lopreiato



Your Workshop TODO:

- 1.** Put away your laptop (until we say so) and turn off your phone
- 2.** Follow along!
- 3.** Ask lots of questions
- 4.** When it's time to try things out, **READ OUR DOCS.**

[illegible]

Resources + Documents for this workshop

1. [Unix/Git Workflow Cheatsheet](#)
2. [Git Cheatsheet](#)
3. [Workshop Repository + README](#)
4. [Presentation Slides](#)



1337 h4ck3rz??

- Why 1337 h4ck1ng?
- Why Git?
- Why Linux?

...well, you know why
hacking, but git and
linux are the essential
tools of good programming

[illegible]

Goals of this workshop:

1. you know the basic tools to be a good CS and software developer
2. *You feel like a like 1337 h4ck3r*
3. You have a jumping off point to build up a digital portfolio and dive into software projects



This workshop

PART I: Linux and UNIX-like environments

- A. Linux background, the command line, + commands
- B. Command line hacking: make a directory!
- C. More background!
- D. More Command line hacking: edit a file in vim!

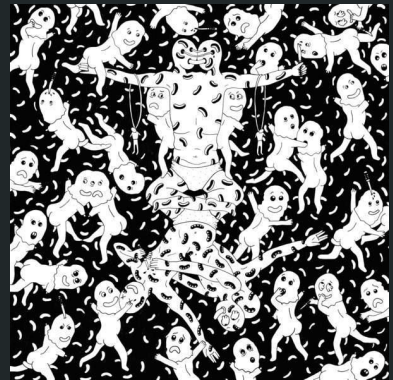
PART II: Git and version control

- A. git background
- B. get git; try out github
- C. github background
- D. Clone our git repo + add a file; make your first pull request

PART I: Linux and UNIX-like environments

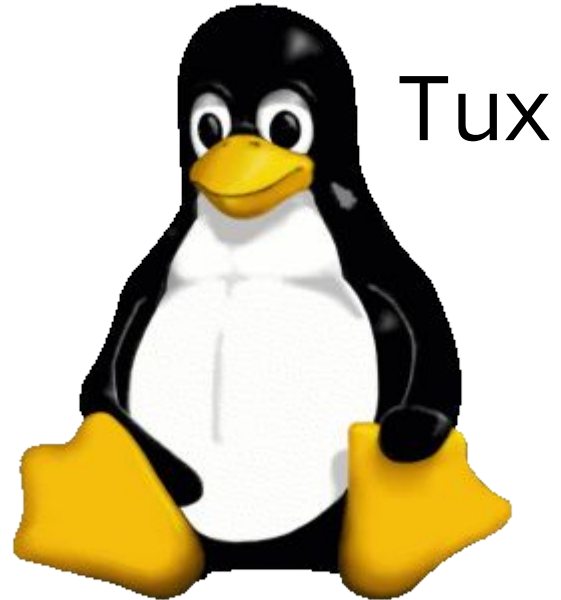
*“In thin socks, I'm butthurt...**crashing Linux server**”*

- Busdriver, “Worlds To Run (ft. Anderson.Paak & milo)”



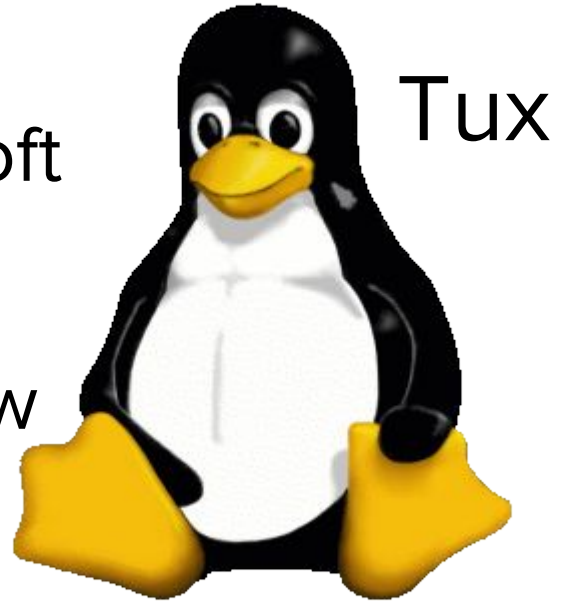
What is a Linux.

- An Open Source Operating System modelled on UNIX.
- Always FREE and community built & maintained.
- All supercomputers use linux and most servers.



History of Linux.

- Released on September 17th, 1991 by Linus Torvald
- Took off fast to escape Microsoft monopoly
- Many different distributions now available



What is a command line interface

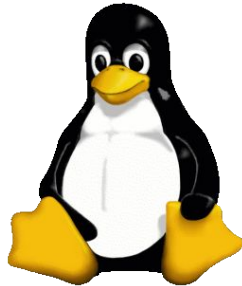
- a text-based application for viewing, handling, and manipulating files directly on your computer
- What 1337 h4ck3rz/real CS use

```
jim - bash -- 102x19
Last login: Tue Sep 25 12:52:00 on console
Jim-Hoskinss-iMac:~ jim$ ls -l ~
total 0
drwx-----+ 7 jim staff 238 Sep 25 12:57 Desktop
drwx-----+ 4 jim staff 136 Sep 25 12:50 Documents
drwx-----+ 4 jim staff 136 Sep 25 12:50 Downloads
drwx-----@ 31 jim staff 1054 Sep 25 12:53 Library
drwx-----+ 3 jim staff 102 Sep 25 12:50 Movies
drwx-----+ 3 jim staff 102 Sep 25 12:50 Music
drwx-----+ 4 jim staff 136 Sep 25 12:50 Pictures
drwxr-xr-x+ 5 jim staff 170 Sep 25 12:50 Public
Jim-Hoskinss-iMac:~ jim$
```

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING test.gripe.wikipedia.org (208.49.152.2) 56(84) bytes of data:
^C
^C
64 bytes from test.gripe.wikipedia.org: ping statistics:
  0 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost ~]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x. 2 root root 4096 May 14 00:15 account
drwxr-xr-x. 1 root root 4096 Jul 31 22:25 cache
drwxr-xr-x. 3 root root 4096 May 18 16:03 db
drwxr-xr-x. 3 root root 4096 May 18 16:03 empty
drwxr-xr-x. 2 root root 4096 May 18 16:03 execs
drwxr-xr-x. 1 2 root gdm 4096 Jun 2 18:39 gdm
drwxr-xr-x. 20 root root 4096 May 18 16:03 lib
drwxr-xr-x. 2 root root 4096 May 18 16:03 local
lrwxrwxrwx. 1 root root 11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx. 1 root root 10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x. 2 root root 4096 May 18 16:03 nls
drwxr-xr-x. 2 root root 4096 May 18 16:03 opt
drwxr-xr-x. 2 root root 4096 May 18 16:03 preserve
drwxr-xr-x. 2 root root 4096 Jul 31 22:11 report
drwxrwxrwx. 1 root root 6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
lrwxrwxrwx. 4 root root 4096 Sep 12 23:54 systemd
drwxr-xr-x. 2 root root 4096 May 18 16:03 yp
[root@localhost ~]# find /usr -name *.rpm
/usr/share/doc/.../langpacks-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../presto-0.8.0-1.el6.noarch.rpm
/usr/share/doc/.../refresh-packagekit-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../remove-with-leaves-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../confusion-free-updates-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../confusion-free-updates-primary_db-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../confusion-mot-free-updates-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../updates/metalink-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../updates-1.0.0-1.el6.noarch.rpm
/usr/share/doc/.../updates-primary_db-1.0.0-1.el6.noarch.rpm
73% [=====] 62 kB/s 2.6 MB 00:15 ETA
```

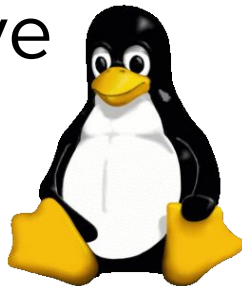
Basic Linux commands

ls	-	list files
cd	-	change directory
mv	-	move a file
mkdir	-	make directory
rm	-	delete a file



basic Linux names/commands, cont.

- > **[filename]** - make a new file
- ~ - reference to the home directory
- - reference to this (current) directory
- .. - reference to directory above current directory



It's hacking time! (part I)

Do now (on your laptop):

1. open <https://github.com/gw-acm/git-linux-2017> for UNIX cheat sheet, basic instructions, + links
2. <https://acm.seas.gwu.edu/ws/git-linux-17/index.html>
3. **Then, - Mac/Unix:** open Terminal (command line)
 - **Windows:** download **git bash** and **open it**

Do now, cont: (on the command line):

1. list visible directories from current position
 2. Make a directory
 3. Navigate into that directory
-
4. make a file called 'new.txt'
 5. Copy a file from another location and put it in this folder (.)
 6. Remove that file

DEMO I!

More complex Linux commands

pwd

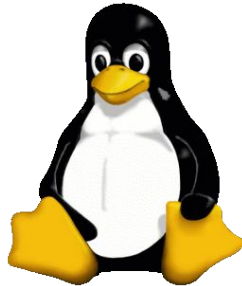
- print working directory

head [file]

- show first 10 lines of file

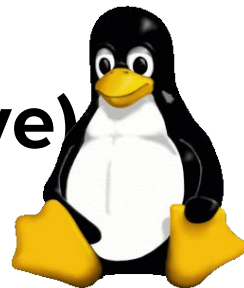
tail [file]

- show last 10 lines of file



Vim commands

- vim [filename]** - open [filename] with vim
- esc** - go back to normal mode
- i** - go to insert
- :wq** - write (save) and quit
- :w** - write
- :q!** - force quit (doesn't save)



It's hacking time! (part II)

Do now (on the command line):

1. enter 'cd'
2. Navigate back to directory
3. enter **vim** and edit 'new.txt'
4. In vim, edit that file
5. Write and exit vim
6. Look at the first lines of that file
7. Remove that file

DEMO II!

PART II: GIT and version control

Git your life together

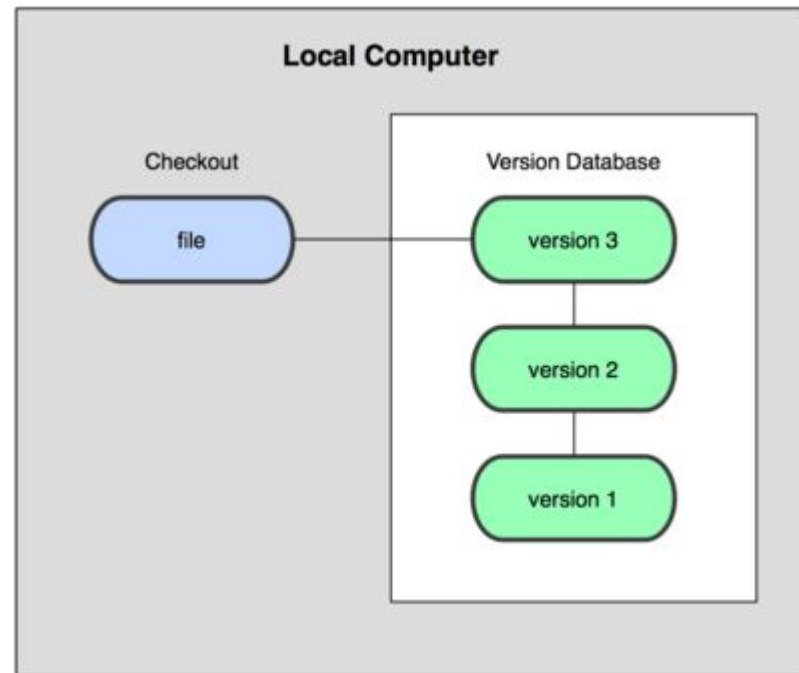
Slides adapted from Neel Shah and Phil Lopreiato





What is git?

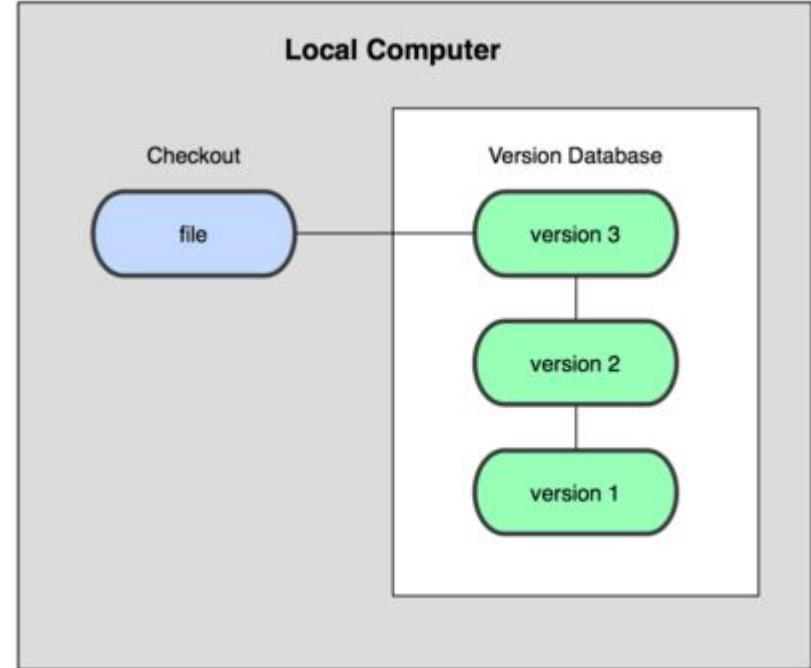
- Git is a type of *version control software*
 - It's a way to track changes to files
- How would you do that?
 - Many zip files
 - Timestamped directories
 - A specially ordered collection of stones
 - Magic?
- Somebody smart decided to synchronize their files with a local database - this is **local version control**





How is git structured?

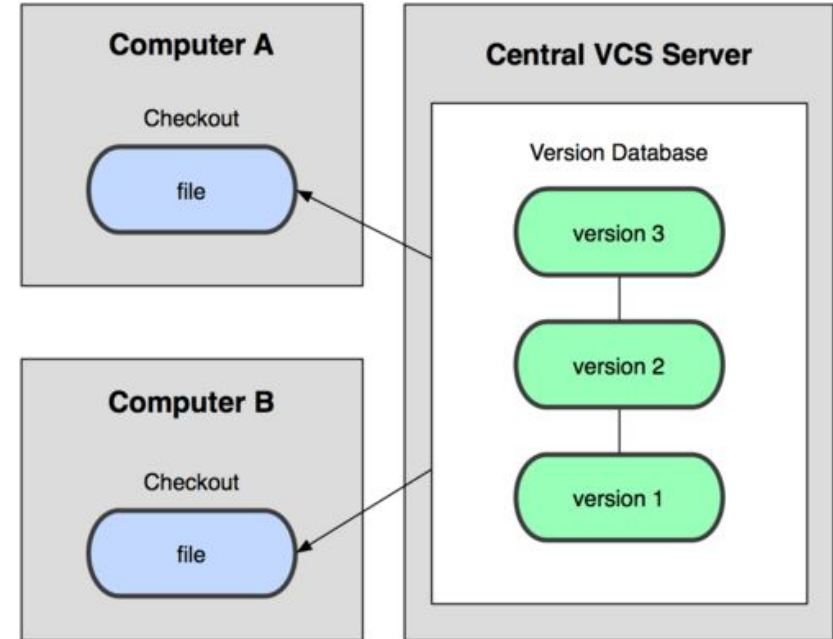
- You keep projects in **repositories** (repos) that are backed up on a server
- you work in local copies of repos, then push changes to server





What Is It?

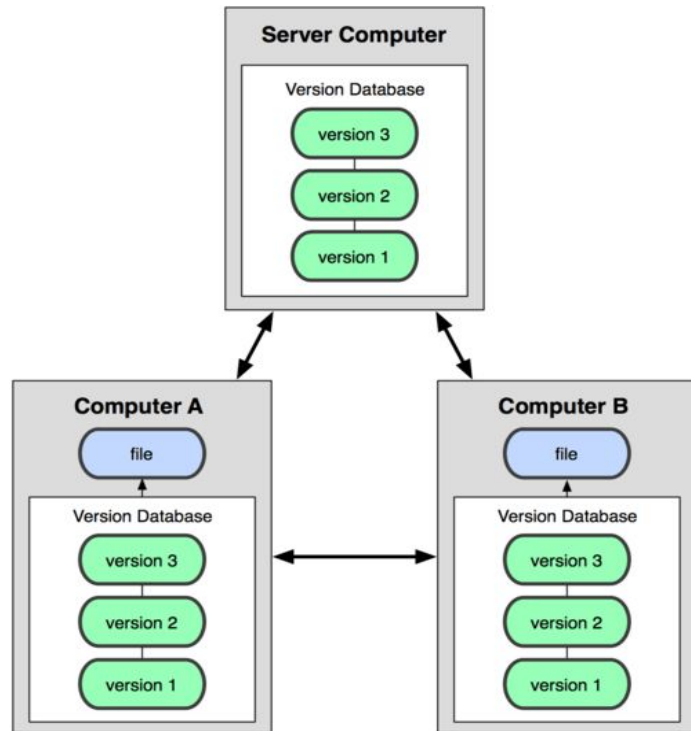
- But what if you want to share your changes with someone else?
 - Easy - just send the database to a server!
- This is called **centralized version control**
- But what happens if the central server goes down?



What Is It?

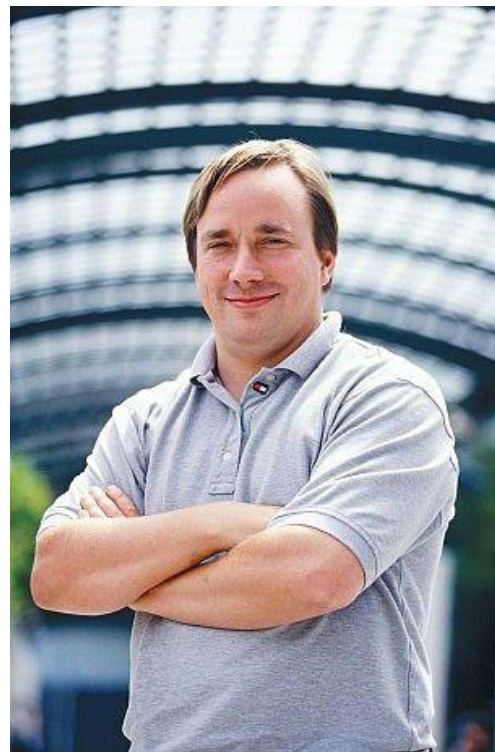


- Just make clients check out the full contents of the repository then!
 - Now, everybody who works on a project has a full copy of the history in case something happens
- This is called **distributed version control**
 - Many common VCS systems use this (git or Mercurial, for example)
- **GitHub** (github.com) is a popular web-based Git repository hosting service



Git Ancient History

- In 2005, the Linux Kernel project needed a new source control system
- Linus Torvalds set out to write his own
 - Popular version control software at the time was not “good enough” for him
 - Needed to be distributed and protect against corruption
- Development began April 3, the project was announced April 6, became self-hosting on April 7, and used in the kernel by June



How does it work?

- Git is a **Directed Acyclic Graph** of repository “snapshots”
- Every change is initially done locally
- Every change has verified integrity
 - The repository is “checksummed” after every change
 - That checksum is used to refer to each **commit**

“In many ways you can just see Git as a filesystem — it is content-addressable, and it has a notion of versioning, but I really really designed it coming at the problem from the viewpoint of a filesystem person (hey, kernels is what I do), and I actually have absolutely zero interest in creating a traditional SCM system.” - Linus

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAHAHAHAHAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

It's (git) hacking time!
(part III)

How can I “Git” Git?



- <http://git-scm.com/downloads>
- You can now use git in a terminal or “Git Bash” on

Windows
Configuration...

```
$ git config --global user.name "YOUR NAME"
```

```
$ git config --global user.email "YOUR EMAIL ADDRESS"
```

Do now: make an account!

- Go to <https://github.com/join> and create an account (or log in)
- Get familiar with git!
 - <https://try.github.io/>

GitHub



(Demo unnecessary)

What's this GitHub?

- GitHub makes coding “social” by providing a Git repository hosting service that maintains all of the “distributed” features of Git and adds a social aspect
 - Issue Trackers
 - Code Releases
 - Project Websites
 - Anyone can contribute!



GitHub

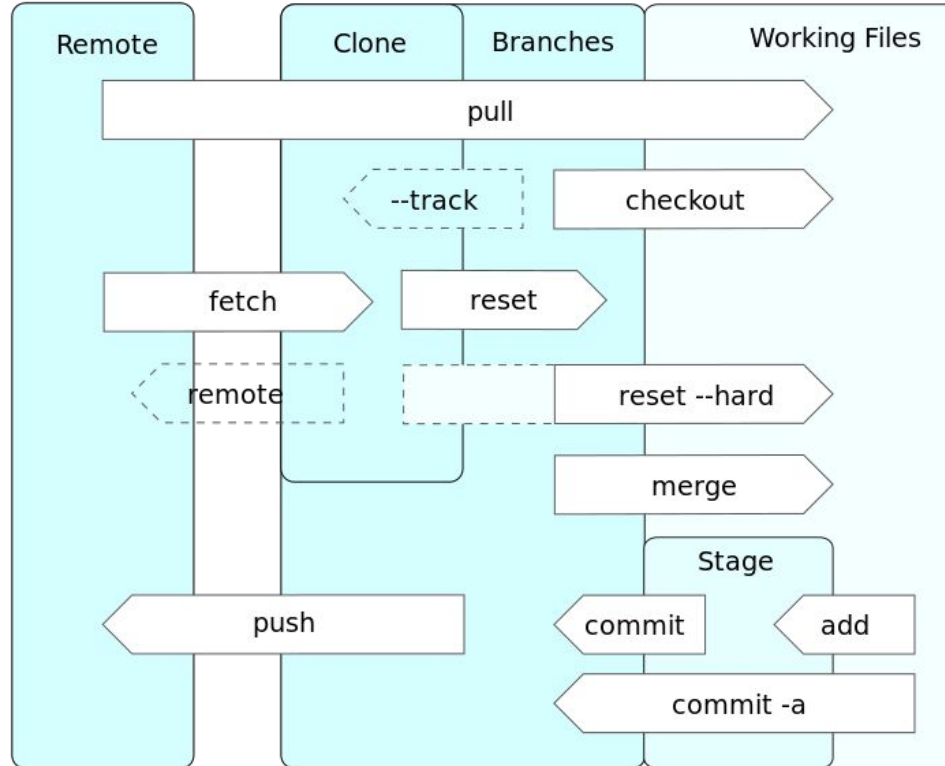
- Example GitHub profiles:
 - <https://github.com/aaroncoplan>
 - <https://github.com/samsaranc>
- Example Repositories:
 - [Linux](#)
 - [The Blue Alliance](#)
- Example Organization:
 - [GWCloudLab](#)
- Explore other repos and projects:
 - <https://github.com/explore>



basic git workflow commands

- git clone “repo url”** - create local copy of repo on your computer
- git add *** - add all files to commit
- git commit -m “init”** - save all your edits with message “init”
- git push** - push your changes to origin branch of repo you’re working on

Distributed versions, visualized



Cloning, forking, and branches



It's (github) hacking
time! (part IV)

Do now (on github + command line):

1. Full instructions in the README:
<https://github.com/gw-acm/git-linux-2017>
2. fork the above repository on your account
3. On the command line, navigate into the directory you made
4. Clone your fork of the repository on your computer
5. Navigate into that repository
6. Use vim to make a .txt file with your name on it (ex: samsara.txt)

Do now (on github + command line):

7. Tell git you made changes with add
8. Commit those changes with a nice message
9. Push them to your forked repo
10. On github, submit a pull request to our original repository

DEMO IV!

Links + Further reading

- Git documentation
 - <https://git-scm.com/docs>
 - <http://wildlyinaccurate.com/a-hackers-guide-to-git/>
- Git + GitHub workflow
 - <https://guides.github.com/activities/hello-world/>
 - <http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1>
 - <http://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html>
- Some other SCMs
 - [Mercurial](#)
 - [SubVersion](#)
- Some other Git hosts
 - [Phabricator](#)
 - [Bitbucket](#)
 - [GitLab](#)

Fin.

Happy h4ck1ng!

